Australian Government
Department of Defence
Defence Science and
Technology Organisation

Optimal 3D Localisation from Image Pixel Measurements

Jijoong Kim and Hatem Hmam[1]

Weapons Systems Division
[1]Electronic Warfare and Radar Division
Defence Science and Technology Organisation

DSTO-TR-2430

ABSTRACT

This report presents a convex relaxation method that globally solves for the camera position and orientation from a set of image pixel measurements associated with a scene of reference points of known 3D positions. The pose optimisation is formulated as a semidefinite positive relaxation program and this approach shows superior performance over existing methods.

RELEASE LIMITATION

Approved for public release

Published by

Weapons Systems Division
DSTO Defence Science and Technology Organisation
PO Box 1500
Edinburgh South Australia 5111 Australia

Telephone: (08) 7389 5555
Fax: (08) 7389 6567

AR-014-794
July 2010

Optimal 3D Localisation from Image Pixel Measurements

Executive Summary

Optical sensors, such as digital cameras, use pixel measurements to estimate the object viewing direction with high accuracy. They are, therefore, prime candidates in defence applications requiring precise localisation and orientation determination. In this report we present a convex relaxation method that globally solves for the camera position and orientation given a set of image pixel measurements associated with a scene of reference points of known 3D positions. The approach formulates the pose optimization problem as a semidefinite positive relaxation (SDR) program. A comprehensive comparative performance analysis, carried out in the computer simulations section, demonstrates the superior performance of our relaxation method over existing approaches. The computational complexity of the method is $O(n)$, where n is the number of reference points.

Authors

Jijoong Kim
Weapons Systems Division

Jijoong Kim received a B.E. (Hons) in Electrical and Electronic Engineering and M.EngSc. both from the University of Adelaide in 1993 and 1995, respectively. He was a research assistant at the EE department of the University of Adelaide for six months, and then joined DSTO in November 1995. During his employment with DSTO, he completed a PhD in Image Processing and Computer Vision from the University of Wollongong in 2006. Since 1995, he has been with the Guidance and Control Group. His research interests include: Missile Guidance, Optimal Control, Navigation, and Computer Vision.

Hatem Hmam
Electronic Warfare and Radar Division

Hatem Hmam received his PhD degree in Electrical Engineering from University of Cincinnati, Ohio, USA, in 1992. He then pursued post-doctoral research in jet engine control at University of Colorado, Boulder, and signal processing at Queensland University of Technology, Brisbane. He joined DSTO in 1996 and worked in WSD and EWRD Divisions. He is currently a member of Distributed Electronic Warfare group where he is developing and implementing passive localisation techniques of RF emitters.

Contents

1. Introduction

The widespread use of calibrated cameras stems from their ability to measure direction or angle (in the form of pixels) with high accuracy. This explains why they are often prime candidates in applications involving precise localisation and orientation determination, particularly in robotics, augmented reality and defence (Campa et al. 2006).

Pose estimation is also known in the literature as the Perspective-n-Point problem (PnP), where the objective is to estimate the camera pose based on image measurements of known reference points (Hartley and Zisserman 2003). The reference points are sometimes called landmarks, control points or markers and their positions are assumed to be given in a world coordinate frame.

The literature on pose estimation is extensive and includes *minimal approaches*, which are restricted to three landmarks (Haralick et al. 1991), and more *generic approaches* that handle more reference points (Ansar and Daniilidis 2003; Lepetit et al. 2008; Lu el at. 2000; Quan and Lan 1999; Schweighofer and Pinz 2006).

Among the latter approaches, some employ iterative techniques (Lu el at. 2000; Schweighofer and Pinz 2006), while others build a large polynomial system and apply linear algebra techniques (Ansar and Daniilidis 2003; Quan and Lan 1999) to solve for the distance information first, then the camera position and rotation. The pose computational complexity varies between $O(n^2)$ (Quan and Lan 1999) and $O(n^8)$ (Ansar and Daniilidis 2003).

Recently a third group of efficient approaches (Lepetit et al. 2008; Schweighofer and Pinz 2008) emerged that are only $O(n)$ and yet display improved performance over many existing pose estimation methods. These will be discussed in some detail in the remaining part of this introduction, but one could also refer to (Lepetit et al. 2008) and the references therein for more details on existing pose approaches.

The *orthogonal iterative* (OI) approach of (Lu el at. 2000) is cited here because of its high estimation accuracy and relatively fast convergence speed provided that the algorithm is properly initialised. Coplanar configuration points or reference points projecting onto one side of the image are known to cause the OI algorithm to perform poorly (Schweighofer and Pinz 2006; Lepetit et al. 2008). A rigorous analysis on the effect of coplanar point scenes is given in (Schweighofer and Pinz 2006), which demonstrates the existence of two ambiguous solutions and proposes a modified algorithm that rectifies the behaviour of the OI algorithm when presented with planar scenes.

The *Efficient PnP* (EPnP) algorithm developed in (Lepetit et al. 2008) uses the concept of virtual control points (VCP) to solve for the camera pose. This concept is novel as it does not directly search for the rotation and translation of the rigidity transformation. It instead solves for the local camera coordinates of four points whose positions are only given in a world reference frame. To achieve this, a constrained quadratic optimisation problem (having 6 quadratic equality constraints) is developed and then approximately solved by performing a four-case null space analysis of a 12 × 12 objective function matrix. Once the local VCP coordinates are estimated the rigid transformation is determined using the absolute orientation algorithm (Horn et al. 1988).

The complexity of this algorithm is $O(n)$ and a number of performance plots were included to demonstrate improved efficiency, particularly in terms of processing speed. The pose estimation accuracy, however, remains lower than that based on minimising the object space error (refer to (Lu el at. 2000) or (Schweighofer and Pinz 2008)). It remains unclear how much the proposed EPnP solving method, which is based on linear algebra techniques, contributes to the observed performance.

The more recent work of (Schweighofer and Pinz 2008) proposes a *sum-of-squares* (SOS) relaxation approach to find the globally optimal solution to the cost function identical to that of (Lu el at. 2000). Its performance accuracy is a significant improvement over those of many previous methods. Its main limitation is that it makes a clear distinction between coplanar and non-coplanar point configurations and therefore executes two variants of the SOS relaxation algorithm before deciding on the globally optimal solution. Execution times are much slower than those of (Lepetit et al. 2008) but are orders of magnitude faster than branch-and-bound based pose estimators (Hartley and Kahl 2009; Olsson et al. 2006; Olsson et al. 2008).

In this report we present a novel *semidefinite program* (SDP) that globally solves for the camera pose. The cost function to be minimised is based on the accumulated object space error as given in (Lu el at. 2000; Schweighofer and Pinz 2008). However, we impose an additional chirality constraint which ensures that the obtained optimal pose is also realistic in the sense that viewed reference points (or at least the majority of them) lie in front of the camera. The computational complexity of our pose estimation method is $O(n)$.

Furthermore, this approach does away with the distinction between coplanar and non-coplanar point configurations, and provides a unified formulation for both configurations. The average recorded runtime is only about 50 ms for a point count of 100, which represents a significant runtime reduction for an algorithm that performs global search for the optimal pose.

This report builds on the work of (Kim and Hmam 2009) and provides a global solution for the pose problem. This report is subdivided as follows: Section 2 gives a formal presentation of the pose problem to be solved. Section 3 presents the proposed solving method based on SDP relaxation. A detailed performance analysis of the method, as applied to synthetic and real data, is carried out in Section 4. Section 5 provides concluding remarks and future research directions.

2. Problem Formulation

Given a set of correspondences between n reference points, $\mathbf{p}_i$, i=1...n in 3D space and their associated images, the objective is to estimate the rotation and position of the calibrated camera. In this work we achieve this estimation through minimising the alignment error sum between the measured and actual pointing vectors as shown below.

$$\begin{aligned}&\underset{\mathbf{R},\mathbf{t}}{\text{minimise}} \sum_{i=1}^{n} w_i \left\| \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|} \times (\mathbf{R}\mathbf{p}_i + \mathbf{t}) \right\|^2 \\ &\text{subject to } \mathbf{R} \in SO(3)\end{aligned} \tag{1}$$

where $\mathbf{v}_i$ is the 3D vector pointing along the i$^{\text{th}}$ measured image pixel and whose origin is the optical centre, $SO(3)$ is the set of 3D rotations, $\mathbf{R}$ is a given rotation from $SO(3)$, and $\mathbf{t}$ represents a 3D translation vector. The parameters, $w_i > 0$, are measurement weights. In this study, they are all set to 1 for simplicity's sake.

Note that the objective function (1) can also be expressed as

$$E(\mathbf{R},\mathbf{t}) = \sum_{i=1}^{n} w_i \left\| \left(\mathbf{I} - \frac{\mathbf{v}_i \mathbf{v}_i^T}{\mathbf{v}_i^T \mathbf{v}_i} \right) (\mathbf{R}\mathbf{p}_i + \mathbf{t}) \right\|^2 \tag{2}$$

which appears in (Lu et al. 2000; Schweighofer and Pinz 2006; Schweighofer and Pinz 2008). For a proof of this refer to Appendix A.

A close examination of the objective function clearly reveals that (1) is minimised when all or most of the cross product vectors, $\frac{\mathbf{v}_i}{\|\mathbf{v}_i\|} \times (\mathbf{R}\mathbf{p}_i + \mathbf{t})$, are small. This occurs if most of the vectors, $\mathbf{R}\mathbf{p}_i + \mathbf{t}$, point approximately along the direction $\mathbf{v}_i$ or in the opposite direction (i.e., $-\mathbf{v}_i$). This second alternative is not acceptable as it implies that the reference points are behind the camera. To enforce this observation, we add the chirality constraint, $\sum_{i=1}^{n} \mathbf{v}_i^T (\mathbf{R}\mathbf{p}_i + \mathbf{t}) \geq 0$, and the optimisation problem then becomes

$$\begin{aligned}&\underset{\mathbf{R},\mathbf{t}}{\text{minimise}}\, E(\mathbf{R},\mathbf{t}) = \sum_{i=1}^{n} w_i \left\| (\mathbf{I} - \mathbf{V}_i)(\mathbf{R}\mathbf{p}_i + \mathbf{t}) \right\|^2 \\ &\text{subject to } \sum_{i=1}^{n} \mathbf{v}_i^T (\mathbf{R}\mathbf{p}_i + \mathbf{t}) \geq 0 \\ &\qquad\qquad \mathbf{R} \in SO(3)\end{aligned} \tag{3}$$

where $\mathbf{V}_i = \frac{\mathbf{v}_i \mathbf{v}_i^T}{\mathbf{v}_i^T \mathbf{v}_i}$.

From (3) the optimal translation vector, $\mathbf{t}_{opt}$, is obtained by setting $\nabla_{\mathbf{t}} E(\mathbf{R},\mathbf{t})$ to zero.

$$\nabla_{\mathbf{t}} E(\mathbf{R},\mathbf{t}) = \sum_{i=1}^{n} w_i \left\{ 2(\mathbf{I}-\mathbf{V}_i)\mathbf{t} + 2(\mathbf{I}-\mathbf{V}_i)\mathbf{R}\mathbf{p}_i \right\} = 0 \qquad (4)$$

Solving (4) leads to

$$\mathbf{t}_{opt}(\mathbf{R}) = -\Lambda^{-1} \sum_{i=1}^{n} \Lambda_i \mathbf{R}\mathbf{p}_i \qquad (5)$$

where $\Lambda = \sum_{i=1}^{n} \Lambda_i$ with $\Lambda_i = w_i(\mathbf{I}-\mathbf{V}_i)$.

The expression for $\mathbf{t}_{opt}$ in equation (5) contains the unknown term $\mathbf{R}$, which needs to be solved. This is done by expressing $\mathbf{R}$ as a function of the coordinates of a unit quaternion (i.e., $\mathbf{q} = [q_1, q_2, q_3, q_4]$).

$$\mathbf{R} = \begin{bmatrix} q_1^2+q_2^2-q_3^2-q_4^2 & 2(q_2q_3+q_1q_4) & 2(q_2q_4-q_1q_3) \\ 2(q_2q_3-q_1q_4) & q_1^2-q_2^2+q_3^2-q_4^2 & 2(q_3q_4+q_1q_2) \\ 2(q_2q_4+q_1q_3) & 2(q_3q_4-q_1q_2) & q_1^2-q_2^2-q_3^2+q_4^2 \end{bmatrix}$$

where $q_1^2+q_2^2+q_3^2+q_4^2=1$.

If we define

$$\mathbf{e} \equiv [q_1^2,\ q_1q_2,\ q_1q_3,\ q_1q_4,\ q_2^2,\ q_2q_3,\ q_2q_4,\ q_3^2,\ q_3q_4,\ q_4^2]^T \qquad (6)$$

then the product term, $\mathbf{R}\mathbf{p}_i$ in (5), can be rearranged as

$$\mathbf{R}\mathbf{p}_i = \mathbf{Q}_i\mathbf{e} \qquad (7)$$

where

$$\mathbf{Q}_i = \begin{bmatrix} p_{ix} & 0 & -2p_{iz} & 2p_{iy} & p_{ix} & 2p_{iy} & 2p_{iz} & -p_{ix} & 0 & -p_{ix} \\ p_{iy} & 2p_{iz} & 0 & -2p_{ix} & -p_{iy} & 2p_{ix} & 0 & p_{iy} & 2p_{iz} & -p_{iy} \\ p_{iz} & -2p_{iy} & 2p_{ix} & 0 & -p_{iz} & 0 & 2p_{ix} & -p_{iz} & 2p_{iy} & p_{iz} \end{bmatrix}$$

and $\mathbf{p}_i = [p_{ix}, p_{iy}, p_{iz}]^T$.

This alternative expression $\mathbf{Q}_i\mathbf{e}$ allows the unknowns in $\mathbf{R}$ to be factored out in the form of linear vectors.

If we let $\mathbf{F} = -\Lambda^{-1}\sum_{i=1}^{n}\Lambda_i\mathbf{Q}_i$ and $\mathbf{G}_i = \mathbf{F} + \mathbf{Q}_i$ then it is straightforward to show that

$$\mathbf{t}_{opt}(\mathbf{R}) = \mathbf{Fe} \tag{8}$$

$$\mathbf{Rp}_i + \mathbf{t}_{opt} = \mathbf{G}_i\mathbf{e}\,. \tag{9}$$

Substituting (9) into the chirality constraint $\sum_{i=1}^{n}\mathbf{v}_i^T(\mathbf{Rp}_i + \mathbf{t}_{opt})$ in (3) leads to

$$\sum_{i=1}^{n}\mathbf{v}_i^T(\mathbf{Rp}_i + \mathbf{t}_{opt}) = \mathbf{h}^T\mathbf{e} \tag{10}$$

where $\mathbf{h} = \sum_{i=1}^{n}\mathbf{G}_i^T\mathbf{v}_i$.

Similarly, the objective function reduces to $E(\mathbf{e}) = \mathbf{e}^T\mathbf{Me}$, where $\mathbf{M} = \sum_{i=1}^{n}\mathbf{G}_i^T\Lambda_i\mathbf{G}_i$ is a 10×10 semidefinite positive matrix. This matrix $\mathbf{M}$ can be computed from the known quantities, $\mathbf{p}_i$ and $\mathbf{v}_i$. Problem (3) then becomes

$$\begin{aligned}&\underset{\mathbf{e}}{\text{minimise}}\ \mathbf{e}^T\mathbf{Me}\\ &\text{subject to } \mathbf{h}^T\mathbf{e} \geq 0\\ &\qquad\qquad \mathbf{k}^T\mathbf{e} = 1\\ &\qquad\qquad \mathbf{e} = [q_1^2,\ q_1q_2,\ q_1q_3,\ q_1q_4,\ q_2^2,\ q_2q_3,\ q_2q_4,\ q_3^2,\ q_3q_4,\ q_4^2]^T\end{aligned} \tag{11}$$

The vector $\mathbf{k} = [1, 0, 0, 0, 1, 0, 0, 1, 0, 1]^T$ ensures that $\mathbf{e}$ satisfies the quaternion normalisation property, $q_1^2 + q_2^2 + q_3^2 + q_4^2 = e_1 + e_5 + e_8 + e_{10} = 1$. Note that before solving (11) it is good practice to normalise $\mathbf{M}$ and $\mathbf{h}$ to improve numerical stability.

3. Semidefinite Programming Relaxation

In this section we apply semidefinite programming theory (Boyd and Vandenberghe 2004) to approximate the solution of (10). The objective function in (11) may be expressed as $\mathbf{e}^T\mathbf{Me} = Tr(\mathbf{Mee}^T) = Tr(\mathbf{MX})$ where $\mathbf{X} = \mathbf{ee}^T$ is rank 1 positive semidefinite and $Tr()$ is the trace operator (i.e., sum of diagonal elements). It can be shown that 20 linear relationships can be established among the entries of $\mathbf{X}$. These relationships originate from the algebraic interdependencies that govern the quartic monomials of $\mathbf{X}$. For example, multiplying the first and fifth elements of $\mathbf{e}$, results in $e_1e_5 = q_1^2q_2^2 = (q_1q_2)^2 = e_2^2$. This then translates into the linear equality constraint, $\mathbf{X}_{15} - \mathbf{X}_{22} = 0$. An exhaustive list of 20 linear component dependencies of $\mathbf{X}$ is given in Appendix B.

The constraint functions in (11) only include quadratic monomials. These constraints can be captured by introducing another rank 1 semidefinite positive matrix, $\mathbf{Y} = \mathbf{qq}^T$, where $\mathbf{q} = [q_1, q_2, q_3, q_4]^T$. Note that, as shown in Appendix B, the trace of $\mathbf{Y}$ is $q_1^2+q_2^2+q_3^2+q_4^2 = \mathbf{k}^T\mathbf{e} = 1$. Furthermore, because any arbitrary entry of $\mathbf{e}$ can also be written as $e_i = e_i(q_1^2+q_2^2+q_3^2+q_4^2)$, then 10 additional constraints may be generated by relating entries of $\mathbf{X}$ to those of $\mathbf{Y}$. For example, given that $e_7 = q_2q_4 (= \mathbf{Y}_{2,4})$, then we have $q_2q_4 = q_1^2q_2q_4 + q_2^3q_4 + q_2q_1^2q_4 + q_2q_4^3$, resulting in the linear equality constraint $\mathbf{X}_{17} + \mathbf{X}_{57} + \mathbf{X}_{87} + \mathbf{X}_{10,7} - \mathbf{Y}_{24} = 0$. Appendix B lists all 10 linear relationships that constrain $\mathbf{X}$ and $\mathbf{Y}$.

Next we combine $\mathbf{X}$ and $\mathbf{Y}$ into a 14×14 block-diagonal matrix $\mathbf{Z} = \begin{bmatrix} \mathbf{X} & \mathbf{0} \\ \mathbf{0} & \mathbf{Y} \end{bmatrix}$. Because both matrices are positive semidefinite, it follows that $\mathbf{Z}$ is also positive semidefinite. Problem (11) can then be relaxed to become

$$\begin{aligned} &\underset{\mathbf{Z}}{\text{minimise}}\ Tr(\overline{\mathbf{M}}\mathbf{Z}) \\ &\text{subject to } Tr(\overline{\mathbf{H}}\mathbf{Z}) \geq 0 \\ &\qquad\qquad Tr(\overline{\mathbf{K}}_i\mathbf{Z}) = b_i,\ 1 \leq i \leq 31 \\ &\qquad\qquad \mathbf{Z} \geq 0 \end{aligned} \tag{12}$$

where $\overline{\mathbf{M}} = \begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$ and $\overline{\mathbf{H}} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{H}} \end{bmatrix}$.

The matrices, $\overline{\mathbf{K}}_i$, $1 \leq i \leq 31$, enforce the 31 equality constraints described above and given in Appendix B. The symmetric matrix, $\tilde{\mathbf{H}}$, is a matrix representation of the vector, $\mathbf{h}$, and is defined such that $\mathbf{h}^T\mathbf{e} = Tr(\tilde{\mathbf{H}}\mathbf{Y})$. Hence

$$\tilde{\mathbf{H}} = \begin{bmatrix} h_1 & h_2 & h_3 & h_4 \\ 0 & h_5 & h_6 & h_7 \\ 0 & 0 & h_8 & h_9 \\ 0 & 0 & 0 & h_{10} \end{bmatrix} \qquad (13)$$

Problem (12) is an SDP relaxation of (11) as the rank 1 constraint on both $\mathbf{X}$ and $\mathbf{Y}$ is dropped. If the ranks of the solutions $\mathbf{X}$ and $\mathbf{Y}$ are both 1, then the global solution of the relaxation problem (12) corresponds to the global solution of (11). If the solutions $\mathbf{X}$ and $\mathbf{Y}$ are not of rank 1, then solving (12) only provides a lower bound for the cost function, $\mathbf{e}^T\mathbf{Me}$, in (11).

It is well known that SDP problems such as (12) can be solved globally in polynomial time using efficient algorithms based on interior point methods (Boyd and Vandenberghe 2004). A number of software packages have been developed and released for public or commercial use (e.g., SeDuMi (Sturm 1999)). Using SeDuMi, for example, the constraints of (12) are captured in a relatively small matrix, $\mathbf{A}$, of size 118×32. This size remains fixed regardless of the number of reference points, n. More details on how to run SeDuMi and how to formulate the input arguments are given in Appendix C.

4. Synthetic and Real Image Results

Before displaying the pose estimation performance plots, we should point out that we have also implemented a fourth degree SOS relaxation algorithm to solve (10). Appendix D provides the details of the SOS relaxation formulation, which bears some similarities with the published algorithm in (Schweighofer and Pinz 2008), except for the additional chirality condition. After executing a large number of SOS-based and SDR pose simulations, we observed almost identical performances in terms of estimation accuracy. However, our SDR algorithm runs about twice as fast (because of the much smaller matrix $\mathbf{A}$).

In this paper we consider a comparative performance analysis with

- the orthogonal iterative (OI) algorithm of (Lu et al. 2000), denoted as LHM,
- the modified OI algorithm of (Schweighofer and Pinz 2006) denoted as MLHM,
- the EPnP algorithm of (Lepetit el al. 2008), denoted as EPnP,
- the proposed SDP relaxation based algorithm, denoted as SDR, and

The SDR algorithm is executed by calling the *sedumi* command (see Appendix C) of the SeDuMi software package (Sturm 1999). The Sedumi accuracy parameter is maintained at 10^{-8}. Accuracies beyond 10^{-8} only result in numerical convergence problems and delays with no significant improvement to the solution accuracy. A local search is optionally applied to refine the obtained SDR solution at a negligible processing cost. This search is implemented as an unconstrained conjugate gradient (CG) descent applied to the cost function of (12) written in terms of Euler angles.

The issue of SDP relaxation tightness has been closely investigated numerically. Relaxation tightness is indicated by a rank 1 solution. However, numerous computer simulations showed decreased relaxation tightness associated with poorer camera / point geometries because the rank 1 condition was frequently not met with strict tolerance. This meant that correction measures were needed to increase the relaxation tightness while preserving convexity. This was achieved by perturbing (12) according to

$$\begin{aligned}
&\min_{\mathbf{Z}} Tr(\overline{\mathbf{M}}\mathbf{Z}) - \alpha\, Tr(\overline{\mathbf{H}}\mathbf{Z}) \\
&\text{subject to } Tr(\overline{\mathbf{H}}\mathbf{Z}) \geq \varepsilon \\
&\qquad\qquad Tr(\overline{\mathbf{K}}_i\mathbf{Z}) = b_i\,,\ 1 \leq i \leq 31 \\
&\qquad\qquad \mathbf{Z} \geq 0
\end{aligned} \tag{14}$$

It was found that setting $\alpha \approx 0.0002$ and $\varepsilon \approx 0.2$, yields significant performance improvement. The main goal of this new formulation is to generate an optimal solution that not only reduces $Tr(\overline{\mathbf{M}}\mathbf{Z})$ but also increases $(\overline{\mathbf{H}}\mathbf{Z})$ beyond ε. Using this new perturbed formulation the SDP relaxation became significantly tighter.

In all computer simulations carried out in this paper, we use a virtual perspective camera model having a focal length of $\mathrm{f} = 800$ and a principal point at $(\mathrm{u}_c, \mathrm{v}_c) = (320, 240)$. The 3D reference points are selected using a uniform distribution within the field of view, and any point that projects outside the 640×480 image plane is discarded from further consideration. Each projected image pixel is then subjected to additive Gaussian noise of varying standard deviations and then quantised by rounding its coordinate to the nearest integer.

We next define the **rotation and position error metrics** as

$$pos\_err = \left\| \mathbf{p}_{true} - \mathbf{p}_{est} \right\| = \left\| \mathbf{R}_{est}^T \mathbf{t}_{est} - \mathbf{R}_{true}^T \mathbf{t}_{true} \right\|$$

$$rot\_err = 2\cos^{-1}(q_0),$$

where $q_0 = 0.5\sqrt{1 + \mathbf{E}_{1,1} + \mathbf{E}_{2,2} + \mathbf{E}_{3,3}}$

and $\mathbf{E}_{1,1}, \mathbf{E}_{2,2}$ and $\mathbf{E}_{3,3}$ are diagonal elements of $\mathbf{E} = \mathbf{R}_{est}\mathbf{R}_{true}^T$.

Finally, we should indicate that all parameters controlling the SDR algorithm are maintained fixed according to pose formulation (14). The associated **A**-matrix in the SeDuMi implementation remains unaffected by the introduced correction terms in (14).

Also, unless otherwise mentioned, the SDR algorithm is assumed to include the fine-tuning step. The Matlab programs for the MLHM and EPnP are obtained from their respective web sources[1]. The EPnP algorithm is always executed with the Gauss-Newton fine-tuning enabled.

4.1 Pose Estimation Error versus Image Pixel Noise

Figure 1 analyses the performance accuracy as a function of image pixel errors. The reference points are selected within the camera field of view, inside the 3D box$[-2, 2] \times [-2, 2] \times [4, 8]$. Fifteen pixel error increments were implemented in the computer simulations.

The four subplots in Figure 1 depict the root mean square (RMS) and median errors of the position and rotation estimates for a 6-landmark configuration selected uniformly within the box. As is clear from the plots, the SDR-based position and rotation estimation accuracies are higher than those of LHM and EPnP. The SDR algorithm followed by local search, labelled as SDR+CG in the figure, display relatively higher position estimation accuracy, particularly for larger pixel errors.

The purpose of including the median estimation error alongside the root mean square error (RMSE) is to mitigate the effect of outliers due to algorithm convergence to local minima. Figure 1 shows comparatively similar RMS and median error plots for the SDP based algorithms but not for the remaining two. This is an indication that significantly more outliers are generated by the EPnP and LHM pose approaches.

Figure 2 is added to closely examine the convergence behaviour of the LHM, EPnP and SDR pose estimators. This figure displays the pose estimation accuracy for a fixed 6-point uncentred configuration. Note that in this and all remaining figures of this paper, we no longer distinguish between SDR+CG and SDR and assume that local fine-tuning forms part of the SDR algorithm. From Figure 2 the LHM performance accuracy is clearly poor even for pixel errors as small as 2. The large deviation of the RMSE error is due to oscillations between two possible poses, one of which presenting a lower minimum. Histograms given in Figure 3 illustrate this point for a

[1] MLHM: http://www.emt.tugraz.at/~pinz/code
EPnP: http://cvlab.epfl.ch/software/EPnP/

pixel measurement error of 5. As many as 550 simulation trials (out of 3000) show the LHM algorithm converging to the wrong pose. Note also how the EPnP estimation error lies within a much larger interval compared to that of the SDR algorithm.

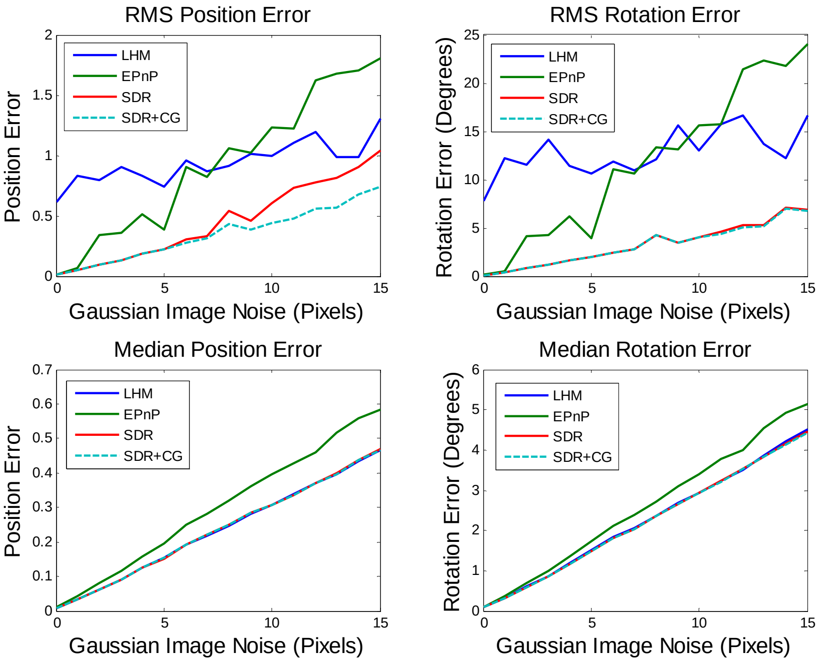

Figure 1: RMS and Median Estimation Errors (Position and Rotation) using 6 uniformly distributed reference points

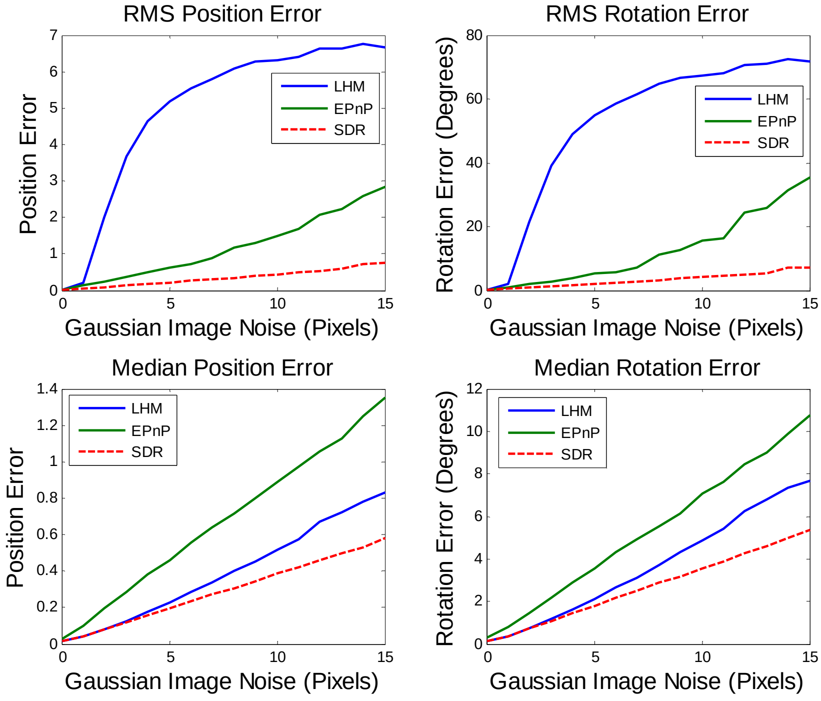

Figure 2: RMS and Median Estimation Errors (Position and Rotation) using 6 fixed reference points

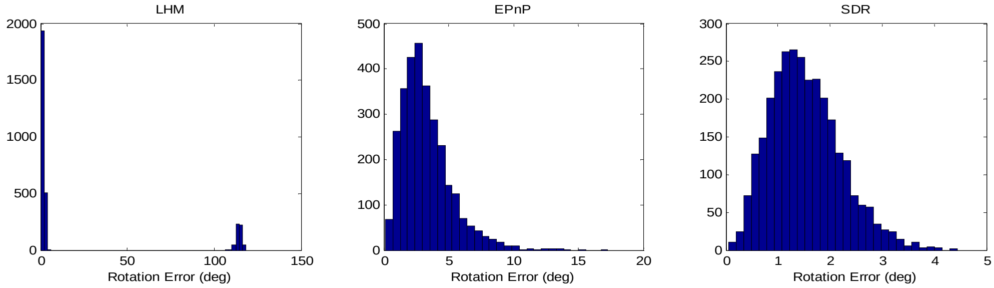

Figure 3: Rotation error histograms for a pixel error of 5 using 3000 realisations.

4.2 Pose Errors versus Number of Reference Points

The next plot in Figure 4 displays the pose (RMSE) as a function of number of reference points. The pixel measurement error is maintained at 5 and the references points are selected randomly within the cube $[-1, 1] \times [-1, 1] \times [5, 7]$. As illustrated in Figure 4 it is expected to observe that the pose estimation accuracy improves for all algorithms as the number of reference points increases. What is less obvious is the considerable error increase for low point counts, particularly when $n = 4$. This behaviour could be explained by the fact that the 4-point pose problem may often be considered as a perturbation of a 3-point pose problem (Haralick el al. 1991), which is known to have as many as 4 solutions, or equivalently 4 global minima. Adding the fourth point often introduces a small disturbance to these minima, making it hard for pose search algorithms to distinguish them. Despite this, Figure 4 clearly shows the superior performance of the SDR algorithm at coping with local minima. This is to be expected, as the SDR formulation (13) is, by design, a close convex approximation of the original nonconvex problem (10).

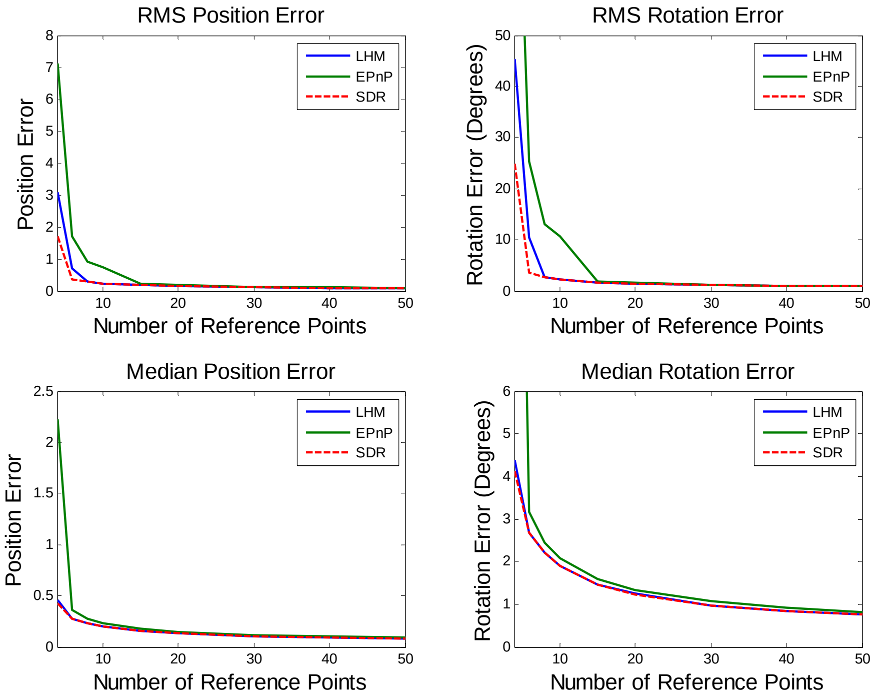

Figure 4: RMS and Median estimation errors (position and rotation) versus number of reference points

4.3 Pose Accuracy Associated With Coplanar Reference Points

Figure 5 depicts the pose accuracy for a coplanar point configuration. We assume a scenario where 10 points lie on a plane pitched by 45° and the objective is to assess the performance accuracy as a function of pixel error. All 10 points are located in the box $[-1,1]\times[-1,1]\times[6,6]$. The two plots on the left of Figure 5 depict the position and rotation accuracy of the MLHM and SDR algorithms when all outliers are removed (i.e., RMS errors are computed after removing outliers). The third plot on the right show the percentage of outliers for MLHM, SDR as well as LHM. As is clear in the figure, both MLHM and SDR exhibit similar behaviours in terms of pose accuracy and percentage of outliers. The LHM algorithm performs poorly even in the absence of pixel measurement errors.

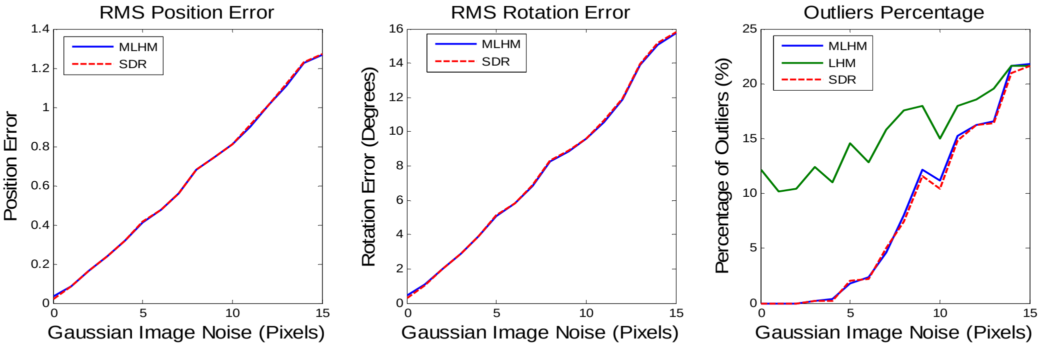

Figure 5: Pose accuracy associated with a coplanar point configuration tilted by 45°. Left: position error. Middle: rotation error. Right: Percentage of pose outliers as a function of pixel error.

4.4 Runtime Comparison

The Matlab generated plots in Figure 6 present the average convergence runtime as a function of number of points for LHM and SDR. The number of points varies from 4 to 1000. For a small number of points the OI algorithm is obviously faster to run. The EPnP algorithm, whose runtime is not shown here, is the fastest (Lepetit et al. 2008). However, one should not lose sight of the fact that other equally important performance indicators exist and need to be considered.

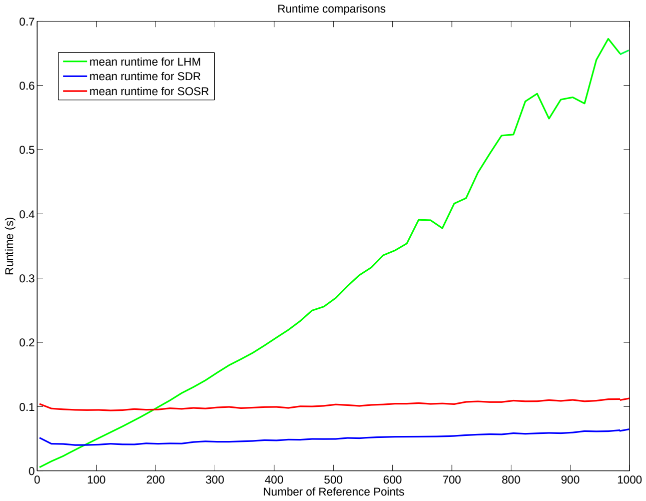

Figure 6: Algorithm runtime comparison plot

In what follows we provide a brief critical summary regarding the overall performance of the LHM, EPnP and SDR pose algorithms, based on numerous computer simulations and observations made in (Lepetit et al. 2008). The LHM algorithm suffers from convergence to local minima unless it is properly initialised. If it converges to the global minimum, then its performance accuracy is better than that of EPnP. The EPnP algorithm does not need initialisation, is less likely to converge to a wrong pose, particularly for smaller pixel errors, but its performance accuracy tends to be lower than that of the LHM algorithm. Both LHM and EPnP algorithms are prone to perform poorly when presented with tilted planar point scenes. The SDR algorithm displays much higher performance accuracy than EPnP and is the least vulnerable to converge to a local minimum. The SDR solution can always be verified for global optimality (by checking the associated SDP solution rank), a feature which is not available for the other two approaches. Its runtime may be perceived as slow for some applications, but only for low point count scenarios (under 50). As the number of points is increased the SDR method becomes significantly faster than the LHM iterative algorithm. To be useful for near-realtime applications, further speedup is recommended. This can be achieved by optimising the algorithm's code and writing an efficient interior point method (Boyd and Vandenberghe 2004) algorithm that solves (13). One should stress that an average runtime of 50 ms for a point count of 100 represents a significant speedup improvement among algorithms that rigorously implement global search for the optimal pose (compare this time with the average runtime of more than a minute using the branch-and-bound search technique of (Hartley and Kahl 2009).

4.5 Pose Estimation Using a Real Image

Figure 7 shows a real image of a hexagonal box taken by a calibrated camera having a focal length of 756 pixels and whose image resolution is 640×480. As many as 10 corners belonging to the box were located in the image and used in the camera pose estimation process by applying the SDR algorithm developed in this work. The estimated camera orientation with respect to the box axes, shown in black in the image, is found to be (roll=31.71°, pitch= 64.42°, yaw=-12.28°). The estimated camera position with respect to the box centre is computed as $[-41.51, 40.8, -23.76]^T$ cm (which corresponds to a translation vector of $\mathbf{t}$=$[-0.18, 2.34, 62.84]^T$ cm). To visualise the pose estimation accuracy, the estimated camera pose parameters are then used to re-project the box wire-frame model on the image as shown in Figure 7.

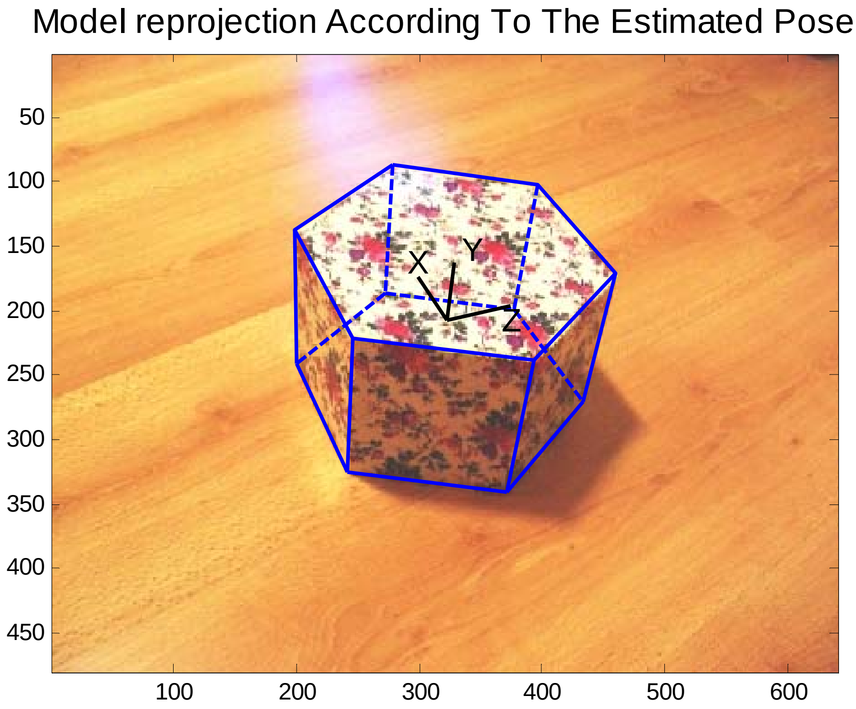

Figure 7: The reprojection of the box model on the original image according to the pose estimated using our SDR algorithm

5. Conclusions

In this paper we have presented an efficient non-iterative pose estimation approach that globally solves for the camera position and orientation. The approach relies on SDP relaxation of a quartic object space error, expressed in terms of quaternion element products. A comprehensive comparative performance analysis, carried out in the computer simulations section, highlights the superior performance of the SDP relaxation based solving method. Its main features are: (1) $O(n)$ computational complexity, (2) improved estimation accuracy for both coplanar and non-coplanar configuration points and (3) relatively high convergence speed, which is reasonable for many time non-critical applications.

6. References

[1] Ansar A. and Daniilidis K. (2003). Linear Pose Estimation from Points or Lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 5, June, (pp 578-589).

[2] Boyd S. and Vandenberghe L. (2004). *Convex Optimisation*, Cambridge University Press.

[3] Campa G., Mammarella, M., Napolitano M. R., Fravolini M. L., Pollini L., and Stolarik B. (2006). A comparison of pose estimation algorithms formachine vision based aerial refueling for UAV, *14th Mediterranean Conference on Control and Automation*, Ancona, Italy, Jun. 28–30, (pp. 1–6).

[4] Grossberg, M. D., and Nayar, S. K., (2001). A general imaging model and a method for finding its parameters. In *IEEE International Conference on Computer Vision*, (pp 1100–1105).

[5] Haralick, R. M., Lee, D., Ottenburg, K., & Nolle, M. (1991). Analysis and solutions of the three point perspective pose estimation problem. *In Conference on computer vision and pattern recognition* (pp. 592–598).

[6] Hartley, R. and Kahl F. (2009). Global Optimisation through Rotation Space Search. *International Journal of Computer Vision*, Vol. 82, (pp 64–79).

[7] Hartley, R. I., & Zisserman, A. (2003). *Multiple view geometry in computer vision* (2nd ed.). Cambridge: Cambridge University Press.

[8] Horn, B. K. P., Hilden, H. M., and Negahdaripour, S. (1988). Closed form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America, 5*(7), (pp 1127–1135).

[9] Kim, J., Hmam and H., (2009). 3D Self-Localisation from Angle of Arrival Measurements. *DSTO-TR-2278*.

[10] Lepetit, V., Moreno-Noguer, F., and Fua, P. (2008). EPnP: An Accurate O(*n*) Solution to the PnP Problem. *International Journal of Computer Vision*.

[11] Lu, C., Hager, G.D., and Mjolsness, E. (2000). Fast and Globally Convergent Pose Estimation from Video Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 6, June, (pp 610-622).

[12] Olsson, C., Kahl, F., and Oskarsson, M. (2006). Optimal estimation of perspective camera pose. *International conference on pattern recognition*, Hong Kong, China.

[13] Olsson, C., Kahl, F., and Oskarsson, M. (2008, in press). Branch and bound methods for Euclidean registration problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*

[14] Prajna, S., Papachristodoulou, A., Seiler, P. and Parrilo, P. A (2004). SOSTOOLS: Sum of squares optimisation toolbox for MATLAB.

[15] Quan, L., and Lan, Z. (1999). Linear *N*-point camera pose determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 21*(7), (pp 774–780).

[16] Schweighofer, G., and Pinz, A. (2006). Robust pose estimation from a planar target. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(12), (pp 2024–2030).

[17] Schweighofer, G. and Pinz, A. (2008). Globally Optimal O(n) Solution to the PnP Problem for General Camera Models. *British Machine Vision Conference.*

[18] Sturm, J. F. (1999). Using SeDuMi 1.02, a Matlab toolbox for optimisation over symmetric cones. *Optimisation Methods and Software*, *11–12*, (pp 625–653).

Appendix A: Proof of Equivalence

To prove that (1) is equivalent to (2), we make use of the cross product property, $\|a \times b\|^2 = \|a\|^2 \|b\|^2 - (a^T b)^2$, which leads to

$$\begin{aligned} E(\mathbf{R},\mathbf{t}) &= \sum_{i=1}^{n} w_i \left[\|\mathbf{Rp}_i + \mathbf{t}\|^2 - \frac{1}{\mathbf{v}_i^T \mathbf{v}_i} \left(\mathbf{v}_i^T (\mathbf{Rp}_i + \mathbf{t}) \right)^2 \right] \\ &= \sum_{i=1}^{n} w_i \left[(\mathbf{Rp}_i + \mathbf{t})^T (\mathbf{Rp}_i + \mathbf{t}) - \frac{1}{\mathbf{v}_i^T \mathbf{v}_i} \left((\mathbf{Rp}_i + \mathbf{t})^T \mathbf{v}_i \right) \left(\mathbf{v}_i^T (\mathbf{Rp}_i + \mathbf{t}) \right) \right] \end{aligned} .$$

By factoring out $\mathbf{Rp}_i + \mathbf{t}$, this expression reduces to

$E(\mathbf{R},\mathbf{t}) = \sum_{i=1}^{n} w_i (\mathbf{Rp}_i + \mathbf{t})^T \left(\mathbf{I} - \frac{\mathbf{v}_i \mathbf{v}_i^T}{\mathbf{v}_i^T \mathbf{v}_i} \right) (\mathbf{Rp}_i + \mathbf{t})$. Since the projection matrix $\mathbf{I} - \frac{\mathbf{v}_i \mathbf{v}_i^T}{\mathbf{v}_i^T \mathbf{v}_i}$ satisfies $\left(\mathbf{I} - \frac{\mathbf{v}_i \mathbf{v}_i^T}{\mathbf{v}_i^T \mathbf{v}_i} \right) = \left(\mathbf{I} - \frac{\mathbf{v}_i \mathbf{v}_i^T}{\mathbf{v}_i^T \mathbf{v}_i} \right)^T \left(\mathbf{I} - \frac{\mathbf{v}_i \mathbf{v}_i^T}{\mathbf{v}_i^T \mathbf{v}_i} \right)$, then (2) follows as a result.

Appendix B: System Matrices and Linear Constraints

$$\mathbf{Z} = \begin{bmatrix} \mathbf{X} & \mathbf{0} \\ \mathbf{0} & \mathbf{Y} \end{bmatrix}_{14\times 14} \quad \text{where}$$

$$\mathbf{X} = \begin{bmatrix}
q_1^4 & q_1^3 q_2 & q_1^3 q_3 & q_1^3 q_4 & q_1^2 q_2^2 & q_1^2 q_2 q_3 & q_1^2 q_2 q_4 & q_1^2 q_3^2 & q_1^2 q_3 q_4 & q_1^2 q_4^2 \\
q_1^3 q_2 & q_1^2 q_2^2 & q_1^2 q_2 q_3 & q_1^2 q_2 q_4 & q_2^3 q_1 & q_2^2 q_1 q_3 & q_2^2 q_1 q_4 & q_3^2 q_1 q_2 & q_1 q_2 q_3 q_4 & q_4^2 q_1 q_2 \\
q_1^3 q_3 & q_1^2 q_2 q_3 & q_1^2 q_3^2 & q_1^2 q_3 q_4 & q_2^2 q_1 q_3 & q_3^2 q_1 q_2 & q_1 q_2 q_3 q_4 & q_3^3 q_1 & q_3^2 q_1 q_4 & q_4^2 q_1 q_3 \\
q_1^3 q_4 & q_1^2 q_2 q_4 & q_1^2 q_3 q_4 & q_1^2 q_4^2 & q_2^2 q_1 q_4 & q_1 q_2 q_3 q_4 & q_4^2 q_1 q_2 & q_3^2 q_1 q_4 & q_4^2 q_1 q_3 & q_4^3 q_1 \\
q_1^2 q_2^2 & q_2^3 q_1 & q_2^2 q_1 q_3 & q_2^2 q_1 q_4 & q_2^4 & q_2^3 q_3 & q_2^3 q_4 & q_2^2 q_3^2 & q_2^2 q_3 q_4 & q_2^2 q_4^2 \\
q_1^2 q_2 q_3 & q_2^2 q_1 q_3 & q_3^2 q_1 q_2 & q_1 q_2 q_3 q_4 & q_2^3 q_3 & q_2^2 q_3^2 & q_2^2 q_3 q_4 & q_3^3 q_2 & q_3^2 q_2 q_4 & q_4^2 q_2 q_3 \\
q_1^2 q_2 q_4 & q_2^2 q_1 q_4 & q_1 q_2 q_3 q_4 & q_4^2 q_1 q_2 & q_2^3 q_4 & q_2^2 q_3 q_4 & q_2^2 q_4^2 & q_3^2 q_2 q_4 & q_4^2 q_2 q_3 & q_4^3 q_2 \\
q_1^2 q_3^2 & q_3^2 q_1 q_2 & q_1 q_3^3 & q_3^2 q_1 q_4 & q_2^2 q_3^2 & q_3^3 q_2 & q_3^2 q_2 q_4 & q_3^4 & q_3^3 q_4 & q_3^2 q_4^2 \\
q_1^2 q_3 q_4 & q_1 q_2 q_3 q_4 & q_3^2 q_1 q_4 & q_4^2 q_1 q_3 & q_2^2 q_3 q_4 & q_3^2 q_2 q_4 & q_4^2 q_2 q_3 & q_3^3 q_4 & q_3^2 q_4^2 & q_4^3 q_3 \\
q_1^2 q_4^2 & q_4^2 q_1 q_2 & q_4^2 q_1 q_3 & q_4^3 q_1 & q_2^2 q_4^2 & q_4^2 q_2 q_3 & q_4^3 q_2 & q_3^2 q_4^2 & q_4^3 q_3 & q_4^4
\end{bmatrix}$$

and

$$\mathbf{Y} = \mathbf{q}\mathbf{q}^T = \begin{bmatrix}
q_1^2 & q_1 q_2 & q_1 q_3 & q_1 q_4 \\
q_1 q_2 & q_2^2 & q_2 q_3 & q_2 q_4 \\
q_1 q_3 & q_2 q_3 & q_3^2 & q_3 q_4 \\
q_1 q_4 & q_2 q_4 & q_3 q_4 & q_4^2
\end{bmatrix} = \begin{bmatrix}
e_1 & e_2 & e_3 & e_4 \\
e_2 & e_5 & e_6 & e_7 \\
e_3 & e_6 & e_8 & e_9 \\
e_4 & e_7 & e_9 & e_{10}
\end{bmatrix}$$

A close examination of the upper triangular region of $\mathbf{X}$ reveals 20 equal entry pairs. These translate into the 20 linear relationships,

$$\mathbf{X}_{4,4} - \mathbf{X}_{1,10} = 0$$
$$\mathbf{X}_{7,7} - \mathbf{X}_{5,10} = 0$$
$$\mathbf{X}_{9,9} - \mathbf{X}_{8,10} = 0$$
$$\mathbf{X}_{2,2} - \mathbf{X}_{1,5} = 0$$
$$\mathbf{X}_{6,6} - \mathbf{X}_{5,8} = 0$$
$$\mathbf{X}_{3,3} - \mathbf{X}_{1,8} = 0$$
$$\mathbf{X}_{4,7} - \mathbf{X}_{2,10} = 0$$
$$\mathbf{X}_{7,9} - \mathbf{X}_{6,10} = 0$$
$$\mathbf{X}_{9,4} - \mathbf{X}_{3,10} = 0$$
$$\mathbf{X}_{2,3} - \mathbf{X}_{1,6} = 0$$
$$\mathbf{X}_{3,5} - \mathbf{X}_{2,6} = 0$$
$$\mathbf{X}_{3,6} - \mathbf{X}_{2,8} = 0$$
$$\mathbf{X}_{3,7} - \mathbf{X}_{2,9} = 0$$
$$\mathbf{X}_{7,8} - \mathbf{X}_{6,9} = 0$$
$$\mathbf{X}_{4,8} - \mathbf{X}_{3,9} = 0$$
$$\mathbf{X}_{6,7} - \mathbf{X}_{5,9} = 0$$
$$\mathbf{X}_{4,5} - \mathbf{X}_{2,7} = 0$$
$$\mathbf{X}_{3,4} - \mathbf{X}_{1,9} = 0$$
$$\mathbf{X}_{2,4} - \mathbf{X}_{1,7} = 0$$
$$\mathbf{X}_{4,6} - \mathbf{X}_{2,9} = 0$$

Using the property $q_i q_j = q_i q_j (q_1^2 + q_2^2 + q_3^2 + q_4^2)$, the following 10 linear constraints are established

$$\mathbf{X}_{1,1} + \mathbf{X}_{2,2} + \mathbf{X}_{3,3} + \mathbf{X}_{4,4} - \mathbf{Y}_{1,1} = 0$$
$$\mathbf{X}_{2,2} + \mathbf{X}_{5,5} + \mathbf{X}_{6,6} + \mathbf{X}_{7,7} - \mathbf{Y}_{2,2} = 0$$
$$\mathbf{X}_{3,3} + \mathbf{X}_{6,6} + \mathbf{X}_{8,8} + \mathbf{X}_{9,9} - \mathbf{Y}_{3,3} = 0$$
$$\mathbf{X}_{4,4} + \mathbf{X}_{7,7} + \mathbf{X}_{9,9} + \mathbf{X}_{10,10} - \mathbf{Y}_{4,4} = 0$$
$$\mathbf{X}_{2,1} + \mathbf{X}_{2,5} + \mathbf{X}_{2,8} + \mathbf{X}_{2,10} - \mathbf{Y}_{1,2} = 0$$
$$\mathbf{X}_{3,1} + \mathbf{X}_{3,5} + \mathbf{X}_{3,8} + \mathbf{X}_{3,10} - \mathbf{Y}_{1,3} = 0$$
$$\mathbf{X}_{4,1} + \mathbf{X}_{4,5} + \mathbf{X}_{4,8} + \mathbf{X}_{4,10} - \mathbf{Y}_{1,4} = 0$$
$$\mathbf{X}_{6,1} + \mathbf{X}_{6,5} + \mathbf{X}_{6,8} + \mathbf{X}_{6,10} - \mathbf{Y}_{2,3} = 0$$
$$\mathbf{X}_{7,1} + \mathbf{X}_{7,5} + \mathbf{X}_{7,8} + \mathbf{X}_{7,10} - \mathbf{Y}_{2,4} = 0$$
$$\mathbf{X}_{9,1} + \mathbf{X}_{9,5} + \mathbf{X}_{9,8} + \mathbf{X}_{9,10} - \mathbf{Y}_{3,4} = 0$$

The normalisation constraint $q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1$ can be expressed as

$$\mathbf{Y}_{1,1} + \mathbf{Y}_{2,2} + \mathbf{Y}_{3,3} + \mathbf{Y}_{4,4} = 1.$$

Appendix C: Implementation of Sedumi Algorithm

$\mathbf{z} = sedumi(\mathbf{A},\mathbf{b},\mathbf{c})$ solves minimise $\mathbf{c}^T\mathbf{z}$ s.t. $\mathbf{A}^T\mathbf{z} = \mathbf{b} \quad \& \, \mathbf{z} \geq \mathbf{0}$

where $\mathbf{c}^T = \left[\left[\mathbf{M}(1:10,1) \cdots \mathbf{M}(1:10,10)\right], \text{zeros}(1,16), 100, -\alpha\right]$ (1x118) (note α =0.002)

$\mathbf{b} = \left[\text{zeros}(1,30), 1, \varepsilon\right]^T$ (32x1) (note ε =0.3)

$\mathbf{z} = \left[\left[\mathbf{X}(1:10,1) \cdots \mathbf{X}(1:10,10)\right], \left[\mathbf{Y}(1:4,1) \cdots \mathbf{Y}(1:4,4)\right], z(117), z(118)\right]^T$ (118x1)

and **A** is given as

$$\mathbf{A} = \begin{bmatrix} \overline{\mathbf{K}}_1(1:10,1) & \cdots & \overline{\mathbf{K}}_{31}(1:10,1) & \mathbf{0}_{10\times1} \\ \overline{\mathbf{K}}_1(1:10,2) & \cdots & \overline{\mathbf{K}}_{31}(1:10,2) & \mathbf{0}_{10\times1} \\ \vdots & \cdots & \vdots & \vdots \\ \overline{\mathbf{K}}_1(1:10,10) & \cdots & \overline{\mathbf{K}}_{31}(1:10,10) & \mathbf{0}_{10\times1} \\ \overline{\mathbf{K}}_1(11:14,11) & \cdots & \overline{\mathbf{K}}_{31}(11:14,11) & \overline{\mathbf{H}}(11:14,11) \\ \overline{\mathbf{K}}_1(11:14,12) & \cdots & \overline{\mathbf{K}}_{31}(11:14,12) & \overline{\mathbf{H}}(11:14,12) \\ \overline{\mathbf{K}}_1(11:14,13) & \cdots & \overline{\mathbf{K}}_{31}(11:14,13) & \overline{\mathbf{H}}(11:14,13) \\ \overline{\mathbf{K}}_1(11:14,14) & \cdots & \overline{\mathbf{K}}_{31}(11:14,14) & \overline{\mathbf{H}}(11:14,14) \\ 0 & \mathbf{0}_{1\times29} & -1 & 0 \\ 0 & \mathbf{0}_{1\times29} & 0 & -1 \end{bmatrix}_{118\times32}$$

In Matlab, the **A** matrix is constructed as shown below.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

A=sparse(zeros([118,32]));

% constraints 1-20:
A(91,1)=1;  A(34,1)=-1;      % constraint 1:      X(1,10)-X(4,4)=0
A(95,2)=1;  A(67,2)=-1;      % constraint 2:      X(5,10)-X(7,7)=0
A(98,3)=1;  A(89,3)=-1;      % constraint 3:      X(8,10)-X(9,9)=0
A(41,4)=1;  A(12,4)=-1;      % constraint 4:      X(1,5)-X(2,2)=0
A(75,5)=1;  A(56,5)=-1;      % constraint 5:      X(5,8)-X(6,6)=0
A(8,6)=1;   A(23,6)=-1;      % constraint 6:      X(8,1)-X(3,3)=0
A(64,7)=1;  A(92,7)=-1;      % constraint 7:      X(4,7)-X(2,10)=0
A(87,8)=1;  A(96,8)=-1;      % constraint 8:      X(7,9)-X(6,10)=0
A(39,9)=1;  A(93,9)=-1;      % constraint 9:      X(9,4)-X(3,10)=0
A(22,10)=1; A(51,10)=-1;     % constraint 10:     X(2,3)-X(1,6)=0
A(52,11)=1; A(25,11)=-1;     % constraint 11:     X(2,6)-X(5,3)=0
A(53,12)=1; A(18,12)=-1;     % constraint 12:     X(3,6)-X(8,2)=0
A(82,13)=1; A(63,13)=-1;     % constraint 13:     X(2,9)-X(3,7)=0
A(86,14)=1; A(68,14)=-1;     % constraint 14:     X(6,9)-X(8,7)=0
A(83,15)=1; A(38,15)=-1;     % constraint 15:     X(3,9)-X(8,4)=0
A(66,16)=1; A(85,16)=-1;     % constraint 16:     X(6,7)-X(5,9)=0
A(62,17)=1; A(35,17)=-1;     % constraint 17:     X(2,7)-X(5,4)=0
A(33,18)=1; A(81,18)=-1;     % constraint 18:     X(3,4)-X(1,9)=0
A(32,19)=1; A(61,19)=-1;     % constraint 19:     X(2,4)-X(1,7)=0
A(82,20)=1; A(54,20)=-1;     % constraint 20:     X(2,9)-X(4,6)=0
```

```
% constraint 21: X(1,1)+X(2,2)+X(3,3)+X(4,4)-Y(1,1)=0
% constraint 22: X(2,2)+X(5,5)+X(6,6)+X(7,7)-Y(2,2)=0
% constraint 23: X(3,3)+X(6,6)+X(8,8)+X(9,9)-Y(3,3)=0
% constraint 24: X(4,4)+X(7,7)+X(9,9)+X(10,10)-Y(4,4)=0
% constraint 25: X(2,1)+X(2,5)+X(2,8)+X(2,10)-Y(1,2)=0
% constraint 26: X(2,1)+X(3,5)+X(3,8)+X(3,10)-Y(1,3)=0
% constraint 27: X(4,1)+X(4,5)+X(4,8)+X(4,10)-Y(1,4)=0
% constraint 28: X(6,1)+X(6,5)+X(6,8)+X(6,10)-Y(2,3)=0
% constraint 29: X(7,1)+X(7,5)+X(7,8)+X(7,10)-Y(2,4)=0
% constraint 30: X(9,1)+X(9,5)+X(9,8)+X(9,10)-Y(3,4)=0
% constraint 31: Y(1,1)+Y(2,2)+Y(3,3)+Y(4,4)-z(117)=1[2]

A(1,21)=1;  A(12,21)=1; A(23,21)=1; A(34,21)=1;  A(101,21)=-1; % const 21
A(12,22)=1; A(45,22)=1; A(56,22)=1; A(67,22)=1;  A(106,22)=-1; % const 22
A(23,23)=1; A(56,23)=1; A(78,23)=1; A(89,23)=1;  A(111,23)=-1; % const 23
A(34,24)=1; A(67,24)=1; A(89,24)=1; A(100,24)=1; A(116,24)=-1; % const 24
A(2,25)=1;  A(42,25)=1; A(72,25)=1; A(92,25)=1;  A(105,25)=-1; % const 25
A(3,26)=1;  A(43,26)=1; A(73,26)=1; A(93,26)=1;  A(109,26)=-1; % const 26
A(4,27)=1;  A(44,27)=1; A(74,27)=1; A(94,27)=1;  A(113,27)=-1; % const 27
A(6,28)=1;  A(46,28)=1; A(76,28)=1; A(96,28)=1;  A(110,28)=-1; % const 28
A(7,29)=1;  A(47,29)=1; A(77,29)=1; A(97,29)=1;  A(114,29)=-1; % const 29
A(9,30)=1;  A(49,30)=1; A(79,30)=1; A(99,30)=1;  A(115,30)=-1; % const 30
A(101,31)=1;A(106,31)=1;A(111,31)=1;A(116,31)=1; A(117,31)=-1; % const 31


%constraint 32
%h(1)Y(1,1)+h(2)Y(1,2)+h(3)Y(1,3)+h(4)Y(1,4)+h(5)Y(2,2)+h(6)Y(2,3)+ …
%h(7)Y(2,4)+h(8)Y(3,3)+h(9)*Y(3,4)+h(10)*Y(4,4)-z(118)=0.3

A(101,32)= H(1); A(105,32)= H(2); A(109,32)= H(3);
A(113,32)= H(4); A(106,32)= H(5); A(110,32)= H(6);
A(114,32)= H(7); A(111,32)= H(8); A(115,32)= H(9);
A(116,32)= H(10); A(118,32)=-1;
```

The optimal rotation can be computed by firstly extracting **Y** matrix from the solution **z** and then constructing the rotation matrix **R**.

```
%extract Y matrix from z
Y=mat(z(101:116),4);

%Construct the rotation matrix R
R=[Y(1,1)+Y(2,2)-Y(3,3)-Y(4,4),2*(Y(2,3)+Y(1,4)),2*(Y(2,4)-Y(1,3));...
   2*(Y(2,3)-Y(1,4)),Y(1,1)-Y(2,2)+Y(3,3)-Y(4,4),2*(Y(3,4)+Y(1,2));...
   2*(Y(2,4)+Y(1,3)),2*(Y(3,4)-Y(1,2)),Y(1,1)-Y(2,2)-Y(3,3)+Y(4,4)];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

The optimal translation **t** can be obtained by inserting **R** into Equation (5) in Section 2, and then the camera position is computed as $\mathbf{p}_{\text{camera}} = -\mathbf{R}^T\mathbf{t}$.

The graphical representation of the sparse **A** matrices for SDR and SOSP are shown below for dimensional comparisons.

[2] Constraint 31 is equivalent to Y(1,1)+Y(2,2)+Y(3,3)+Y(4,4) ≥ 1, which speeds up the SeDuMi runtime, and Y(1,1)+Y(2,2)+Y(3,3)+Y(4,4) will be brought very close to 1 at the end of the run.

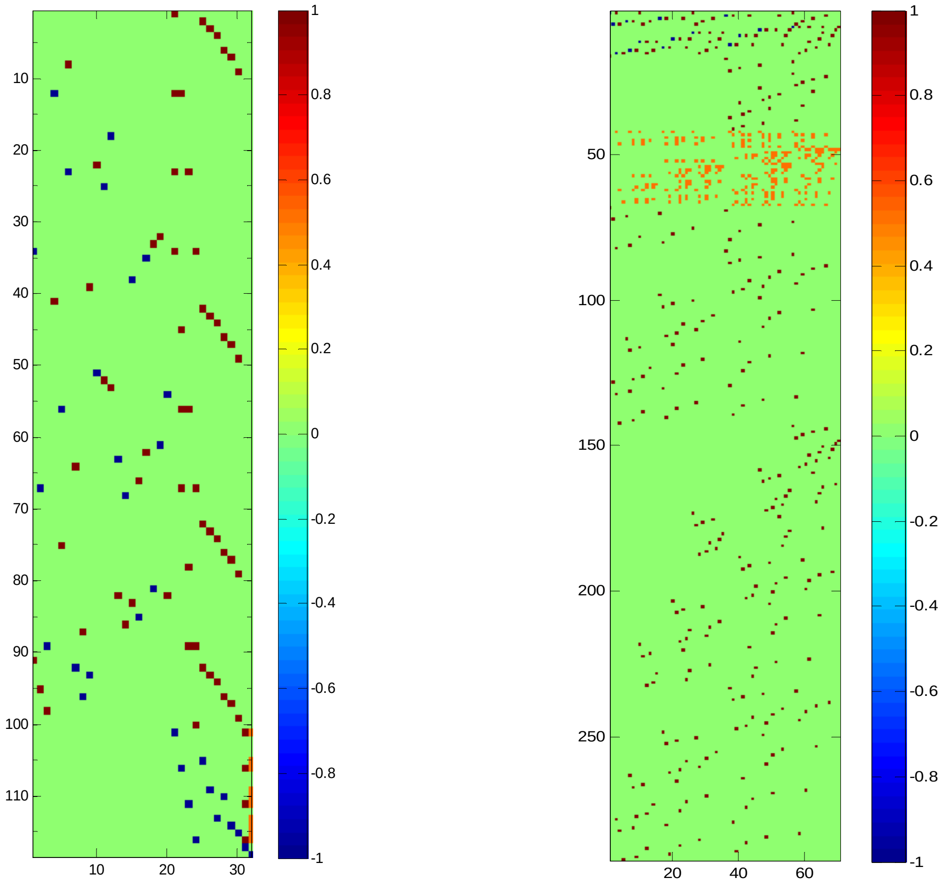

Figure 8: Image representation of A matrices used as input to SeDuMi Software

Appendix D: Sum of Squares Programming

A multivariate polynomial, $p(x)$, of degree $2l$, is a sum of squares (SOS) if it can be expressed as a finite sum of polynomial squares (i.e. $p(x)=\sum p_i^2(x)$). This clearly means that any SOS polynomial can be written as $p(x)=\mathbf{m}^T(x)\mathbf{Q}\mathbf{m}(x)$, where $\mathbf{Q}$ is a semidefinite positive matrix consisting of affine entries and $\mathbf{m}(x)$ is a monomial vector whose largest degree is l. SOS polynomials are necessarily non-negative, but not all non-negative polynomials are SOS.

A global minimiser of the multivariate polynomial, $F(x)$, is obtained by solving

$$\underset{x\in R^m}{\text{minimise}}\, F(x) \tag{D1}$$

which is equivalent to solving

$$\begin{aligned}&\text{maximise } \gamma\\ &\text{subject to } F(x)-\gamma\geq 0\end{aligned} \tag{D2}$$

A lower bound, γ, to (D2) can be obtained by relaxing (D2) as

$$\begin{aligned}&\text{maximise } \gamma\\ &\text{subject to } F(x)-\gamma \text{ is SOS}\end{aligned} \tag{D3}$$

whose solution is computable in polynomial time using semidefinite programming.

Now let us assume we are given additional equality and inequality constraints, leading to the constrained minimisation problem

$$\begin{aligned}&\text{minimise } F(x)\\ &\text{subject to } g_i(x)\geq 0,\ i=1,\cdots,M\\ &\qquad\qquad\ h_j(x)=0,\ j=1,\cdots,N\end{aligned} \tag{D4}$$

The lower bound, γ, for (D4) can be found using the Positivstellensatz-based relaxations

$$F(x)-\gamma=\sigma_o(x)+\sum_j\lambda_j(x)h_j(x)+\sum_i\sigma_i(x)g_i(x)+\sum_{i1,i2}\sigma_{i1,i2}(x)g_{i1}(x)g_{i2}(x)+\cdots \tag{D5}$$

where $\sigma_i(x)$ are SOS polynomials and $\lambda_j(x)$ are polynomials. Similar to the unconstrained problem (D3), maximising γ, subject to SOS relaxation of (D5) can be computed using a semidefinite program for which efficient well-established algorithms (e.g., interior point methods) exist. The lower bound, γ, approaches the global minimum of (D4) as the degree of (D5) is increased. In practice a moderate increase is often enough to result in a highly accurate approximation of the global minimum. When this occurs, the SOS relaxation is said to be tight.

Pose Estimation Using Sum of Squares Programming (SOSP)

In this section we offer a unified approach to solve the pose estimation problem using SOS relaxation irrespective of whether the reference points are coplanar or non-coplanar. The work in (Schweighofer and Pinz 2008) makes this distinction and formulates a number of SOS relaxation algorithms to account for the type of reference points. In what follows we provide a detailed SOS relaxation applicable to the general case based on solving

$$\begin{aligned} &\underset{\mathbf{q}}{\text{minimise}}\ \mathbf{e}^T(\mathbf{q})\mathbf{M}\mathbf{e}(\mathbf{q}) \\ &\text{subject to}\ \mathbf{h}^T\mathbf{e}(\mathbf{q}) \geq 0 \\ &\qquad\qquad \mathbf{k}^T\mathbf{e}(\mathbf{q}) = 1 \\ &\qquad\qquad q_1 \geq 0 \end{aligned} \tag{D6}$$

which is based on (10). This new formulation emphasises that the unknown vector is $\mathbf{q} = [q_1, q_2, q_3, q_4]^T$ and adds another inequality constraint, $q_1 \geq 0$, which serves to distinguish equally valued global minima generated by opposite solutions $\mathbf{q}$ and $-\mathbf{q}$. The objective function is $\mathbf{F}(\mathbf{q}) = \mathbf{e}^T(\mathbf{q})\mathbf{M}\mathbf{e}(\mathbf{q})$ which is a 4-variable quartic polynomial.

If we put $g_1(\mathbf{q}) = \mathbf{h}^T\mathbf{e}(\mathbf{q})$, $g_2(\mathbf{q}) = a$ and $h(\mathbf{q}) = \mathbf{k}^T\mathbf{e}(\mathbf{q}) - 1$, then using the Positivstellensatz and SOS decomposition, a 4-degree SOS relaxation problem may be formulated as

$$\begin{aligned} &\text{maximise}\ \gamma \\ &\text{subject to}\ \mathbf{C}(\mathbf{q})\ \text{is SOS} \end{aligned} \tag{D7}$$

where the expression of $\mathbf{C}(\mathbf{q})$ being given by

$$\mathbf{C}(\mathbf{q}) = \mathbf{F}(\mathbf{q}) - \gamma - \lambda(\mathbf{q})h(\mathbf{q}) - \sigma_1(\mathbf{q})g_1(\mathbf{q}) - \sigma_2(\mathbf{q})g_2(\mathbf{q}) - \sigma_{12}(\mathbf{q})g_1(\mathbf{q})g_2(\mathbf{q})$$
$$\text{with}\ \sigma_1, \sigma_2, \sigma_{12}\ \text{being SOS}$$

The expressions for the introduced parameter polynomials are

$$\begin{aligned} &\sigma_1(\mathbf{q}) = [1;\mathbf{q}]^T\mathbf{Q}_1[1;\mathbf{q}] \\ &\sigma_2(\mathbf{q}) = [1;\mathbf{q}]^T\mathbf{Q}_2[1;\mathbf{q}] \\ &\sigma_{12}(\mathbf{q}) = \sigma\ \text{and} \\ &\lambda(\mathbf{q}) = [1;\mathbf{q}]^T\mathbf{Q}_0[1;\mathbf{q}] \end{aligned}$$

with $\mathbf{Q}_1 \geq 0$, $\mathbf{Q}_2 \geq 0$ and σ a non-negative scalar. The degrees of the SOS polynomials $\sigma_1, \sigma_2, \sigma_{12}$ and the polynomial, λ, are determined by the requirement that the degree of $\mathbf{C}(\mathbf{q})$ is 4 and the property that SOS polynomials are even degree polynomials. The matrix $\mathbf{Q}_0$ is selected to be an upper-triangular matrix with 15 unknown coefficients.

The SOS relaxation of (D7) means that $\mathbf{C}(\mathbf{q})$ can also be written as $\mathbf{C}(\mathbf{q}) = \mathbf{m}^T(\mathbf{q})\mathbf{Q}\mathbf{m}(\mathbf{q})$ where $\mathbf{Q} \geq 0$ and $\mathbf{m}(\mathbf{q}) = [1; \mathbf{q}; \mathbf{e}(\mathbf{q})]$ is a 15-dimensional vector whose largest degree is 2. By expanding and matching both expressions of $\mathbf{C}(\mathbf{q})$, all coefficient matrix entries of $\mathbf{Q}_0$, $\mathbf{Q}_1$, $\mathbf{Q}_2$, $\mathbf{Q}$ and the scalars σ, γ could be related affinely. With the aid of a SOSP software package such as SOSTOOLS (Prajna et al. 2004), the matching process can be automated and the SDP formulation may be established. However, due to the large dimensions of the sparse matrix $\mathbf{A}$ (size: 292×70) and vectors $\mathbf{b}$ and $\mathbf{c}$, these are not included in the paper and are obtainable from the authors. Note that if one wishes to further tighten the SOS relaxation by increasing the degree beyond 4, then the matrix $\mathbf{A}$ becomes inconveniently large, resulting in a significant increase in processing requirement.

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION
DOCUMENT CONTROL DATA

1. PRIVACY MARKING / CAVEAT (OF DOCUMENT)

2. TITLE

Optimal 3D Localisation from Image Pixel Measurements

3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION)

Document	(U)
Title	(U)
Abstract	(U)

4. AUTHOR(S)

Jijoong Kim and Hatem Hmam

5. CORPORATE AUTHOR

DSTO Defence Science and Technology Organisation
PO Box 1500
Edinburgh South Australia 5111 Australia

6a. DSTO NUMBER DSTO-TR-2430	6b. AR NUMBER AR-014-794	6c. TYPE OF REPORT Technical Report	7. DOCUMENT DATE July 2010

8. FILE NUMBER 2010 / 1028475	9. TASK NUMBER 07 / 249	10. TASK SPONSOR CWSD	11. NO. OF PAGES 27	12. NO. OF REFERENCES 18

13. URL on the World Wide Web

http://www.dsto.defence.gov.au/corporate/reports/DSTO-TR-2430.pdf

14. RELEASE AUTHORITY

Chief, Weapons Systems Division

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

Approved for public release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, EDINBURGH, SA 5111

16. DELIBERATE ANNOUNCEMENT

No Limitations

17. CITATION IN OTHER DOCUMENTS Yes

18. DSTO RESEARCH LIBRARY THESAURUS http://web-vic.dsto.defence.gov.au/workareas/library/resources/dsto_thesaurus.shtml

Optimisation, Localisation, Pose Estimation

19. ABSTRACT

This report presents a convex relaxation method that globally solves for the camera position and orientation from a set of image pixel measurements associated with a scene of reference points of known 3D positions. The pose optimisation is formulated as a semidefinite positive relaxation program and this approach shows superior performance over existing methods.